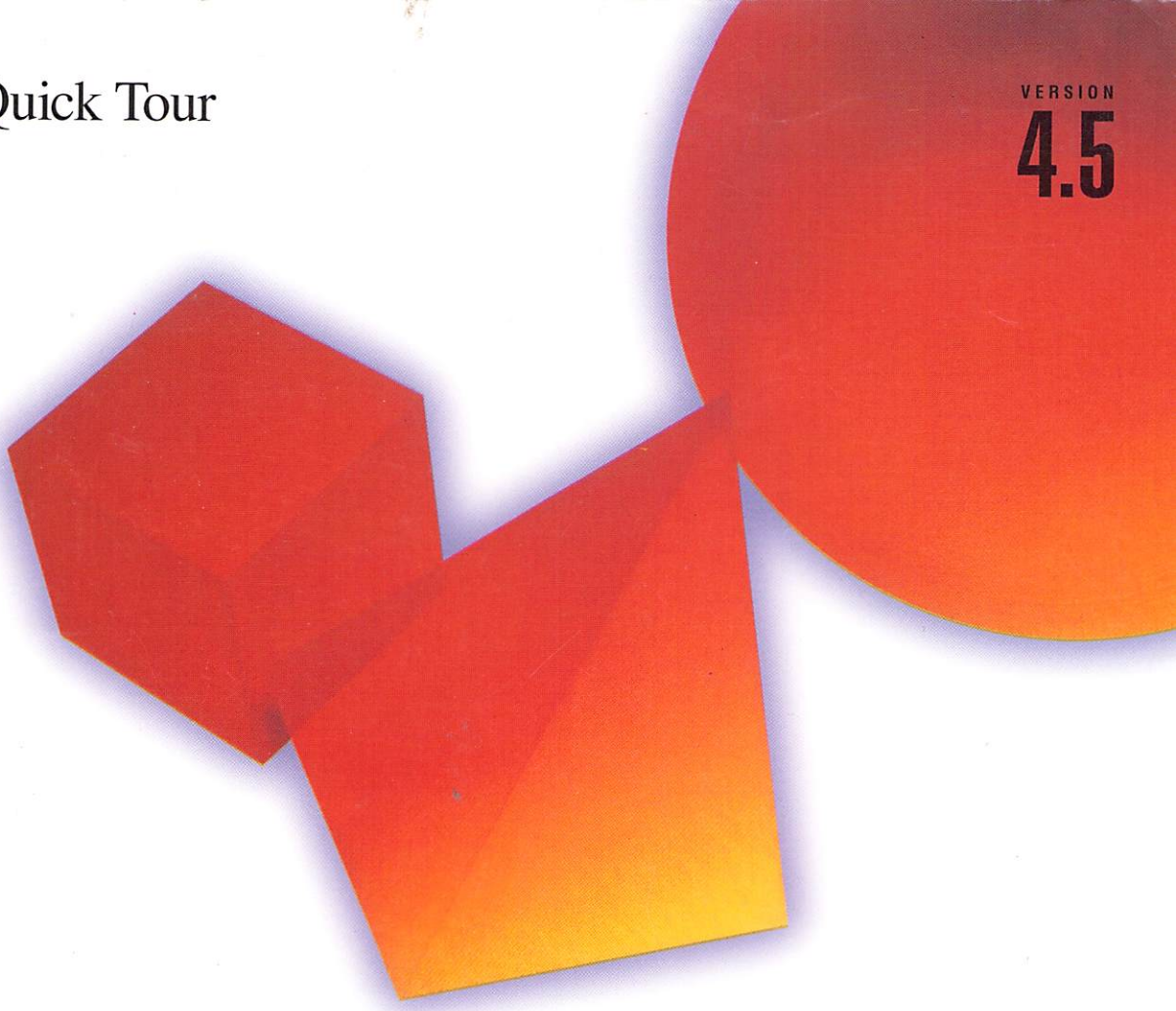
Quick Tour

VERSION 4.5



Borland® C++

A Quick Tour of Borland C++

Introducing Borland C++ *2*

Inside Borland C++ *3*

Building Windows applications *4*

Managing projects *6*

Automating application development *8*

Refining your application *9*

Designing a graphical user interface *10*

Using frameworks and libraries *11*

Debugging and profiling *16*

Building DOS applications *18*

Working from the command line *19*

C and C++ languages *20*

Program portability *22*

Learning more about Borland C++ *24*

Expanding the power of Borland C++ *26*

For more information...

To install Borland C++ on your system, see the instructions in the booklet inside your CD-ROM case.

Introducing Borland C++

Welcome to Borland C++, the most complete C and C++ programming environment available today. With the Borland C++ visual development environment, creating complex applications is as simple as pointing and clicking on rational choices from a menu, or selecting items in a dialog box.

Only Borland C++ meets the latest C and C++ standards while providing innovative support for application development on DOS, Windows 3.x, Win32s, Windows NT, and Windows 95.

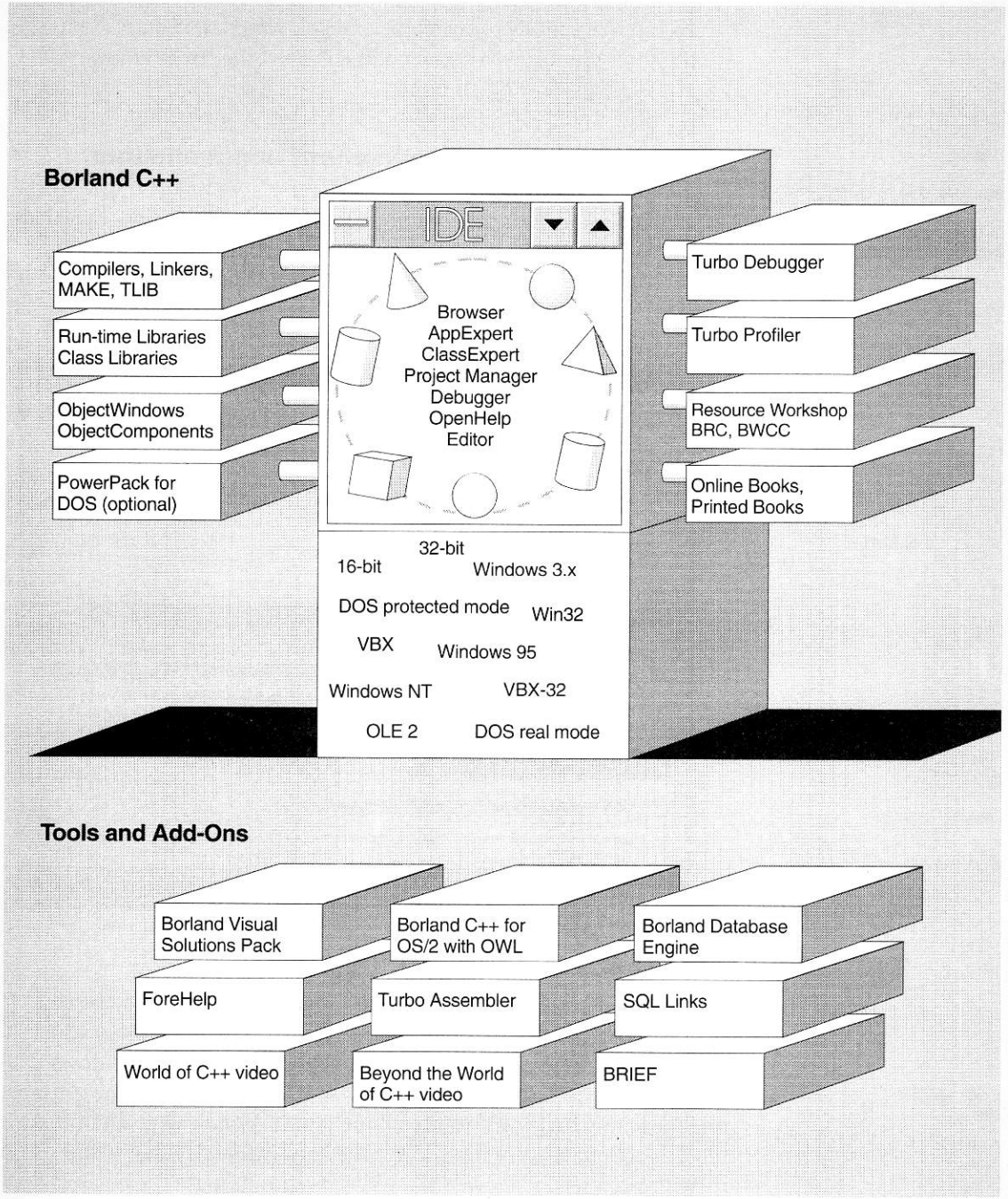
The integrated development environment and tools reduce the amount of work and time required to set up and code your application, allowing you to focus on your work. Whether you are writing new code, maintaining existing applications, debugging, creating Windows resources, or migrating to another PC platform, Borland C++ technology gives you everything you need to work effectively.

With Borland C++, one powerful user interface gives you access to all the tools you need. Use the Project Manager to organize and manage your files. Use AppExpert to create the foundation of your Windows applications, and use ClassExpert to modify and refine these applications. Use Resource Workshop to visually design and create the pieces of your user interface. Use the integrated debugger to step through your code, and Turbo Profiler to view and analyze your program's performance. These are just a few of the many tools Borland C++ provides to make application development easy and efficient.

The Borland C++ class libraries and the ObjectWindows and ObjectComponents frameworks give you access to a powerful collection of classes that solve many of your most difficult Windows programming tasks. ObjectComponents hides the complexity of OLE 2 programming, and gives you the flexibility to create new OLE 2 applications, or simply adds this capability to your existing programs.

The complete documentation set that comes with Borland C++ gives you many ways to access the information you need quickly—through online Help, with online books, and with examples. The full documentation set, provided to all CD-ROM users online with the powerful DynaText viewer, is also available in printed form.

This *Quick Tour* gives you an overview of Borland C++ and its complex feature set. Throughout this booklet, refer to the left margin for guidance on where you can go to find details on the topics that interest you.


Borland C++
IDE
Compilers, Linkers, MAKE, TLIB
Run-time Libraries Class Libraries
ObjectWindows ObjectComponents
PowerPack for DOS (optional)
Browser
AppExpert
ClassExpert
Project Manager
Debugger
OpenHelp
Editor
Turbo Debugger
Turbo Profiler
Resource Workshop BRC, BWCC
Online Books, Printed Books
32-bit
16-bit
Windows 3.x
DOS protected mode
Win32
VBX
Windows 95
Windows NT
VBX-32
OLE 2
DOS real mode
Tools and Add-Ons
Borland Visual Solutions Pack
Borland C++ for OS/2 with OWL
Borland Database Engine
ForeHelp
Turbo Assembler
SQL Links
World of C++ video
Beyond the World of C++ video
BRIEF

For more information...

Building Windows applications

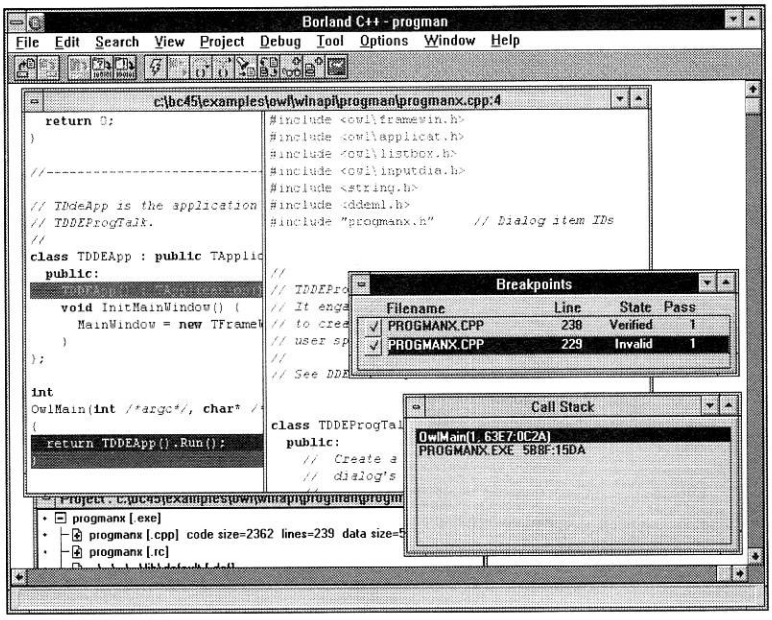
Borland C++ simplifies Windows application development by giving you one environment for all your programming tasks, Experts for quick project starts, example programs for coding ideas, the tools you need to create complete and efficient Windows applications, and much more.

All your program development under one roof

See the *User's Guide*, Part I.

In Help, choose Search All, then Search For "IDE."

THE BORLAND C++ INTEGRATED DEVELOPMENT ENVIRONMENT (IDE) puts all your programming tools within easy reach—you can develop resources, write program code, debug, compile, browse through C++ objects, and reference expert Help without leaving the IDE. You can even customize the IDE so you can call on any special programming tools you need to use.



Fast starts

THE BORLAND C++ EXPERTS and frameworks let you step right into your real programming challenges. Use the Experts so you don't lose time coding the basic functionality of your Windows applications. Use the Frameworks so you don't spend time getting bogged down in the overhead inherent in Windows programming. These Borland C++ tools lower the barrier-to-entry that many engineers associate with Windows programming.

Building Windows applications

The Borland C++ Experts

See the *User's Guide*, Chapters 4 and 5.

The three Experts in Borland C++ give you an instant programming edge. Use TargetExpert to set up your project files. Use AppExpert to avoid mundane Windows programming chores by having it generate the source-code foundation for your projects. Use ClassExpert to customize the code generated by AppExpert.

The ObjectWindows framework

See the *ObjectWindows Programmer's Guide*.

In Help, choose the OWL button.

Also, see "Using frameworks and libraries" in this booklet.

Use the ObjectWindows framework to simplify the complex nature of Windows programming. The numerous high-level objects in ObjectWindows let you focus on solving real problems instead of fighting the complexity often involved with Windows programming. Using the classes in ObjectWindows, you can manage your dialog boxes, scroll bars, bitmaps, and much more. You can even use ObjectWindows to plug VBX controls into your 16- and 32-bit Windows applications.

Productivity tools

See the *User's Guide*, Appendix B.

In Help, choose Search All, then Search For "precompiled headers," "browser," and "editor."

WITH BORLAND C++, YOU CAN USE precompiled headers to compile your programs in a fraction of the time it normally takes. Then, use the Browser to view the hierarchy of your classes. From the Browser, a click of the mouse moves to any file in your project or class.

The Borland C++ editor is fully configurable; you can customize the BRIEF-like editor by changing the key mappings, the syntax highlighting, and the editor SpeedBar. In addition, the editor supports multiple files, and you can view them all simultaneously by splitting the Edit window into multiple panes.

Visual interface design

See "Designing a graphical user interface" in this booklet.

In Help, choose Search All, then Search For "resources."

RESOURCE WORKSHOP IS A VISUAL TOOL that helps you design your user interface by displaying the layout of your UI components as you create them. This helps you make the best use of your time: you don't need to go back and forth between a resource compiler and your application to view your UI.

Using even more tools

See the *User's Guide*, Chapter 2.

In Help, choose Search All, then Search For "tools."

YOU CAN CONFIGURE THE BORLAND C++ IDE to include custom tools, translators, and viewers on the Tools menu. Tools are standalone programs like Turbo Debugger; translators are programs that generate one file type from another; and viewers are programs that let you examine files of a particular type.

For more information...

Managing projects

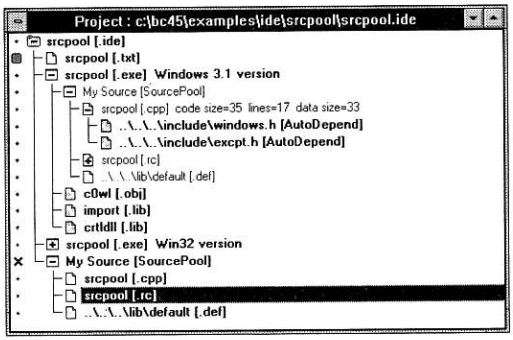
In Help, choose Search All, then Search For "project management."

The Borland C++ Project Manager makes it easy for you to manage your C and C++ projects. Whether your projects are large or small, or if they target Windows, Win32, or DOS, the Project Manager handles them all.

Seeing the whole project

See the *User's Guide*, Chapter 2.

THE PROJECT MANAGER DISPLAYS all your project files in a hierarchy to give you a visual overview of your project organization.



In Help, choose Search All, then Search For "project hierarchy."

You can customize the Project Manager to display only the files you want to see—with the flip of a switch you can view or hide the source-file dependencies (such as header files), the libraries and start-up modules, or the object files contained in your project.

Accessing project source files

In Help, choose Search All, then Search For "editor."

SIMPLY DOUBLE-CLICK A SOURCE-FILE NODE in the Project window to bring that file into an active Edit window. The integrated editor lets you view or modify the contents of any program header file or source file. You can also use the editor to create and modify resource scripts and other ASCII text files.

Creating multipurpose projects

See the *User's Guide*, Chapter 2.

Also, see SRCPOOL.IDE in EXAMPLES\IDE\SRCPOOL.

In Help, choose Search All, then Search For "Source Pools."

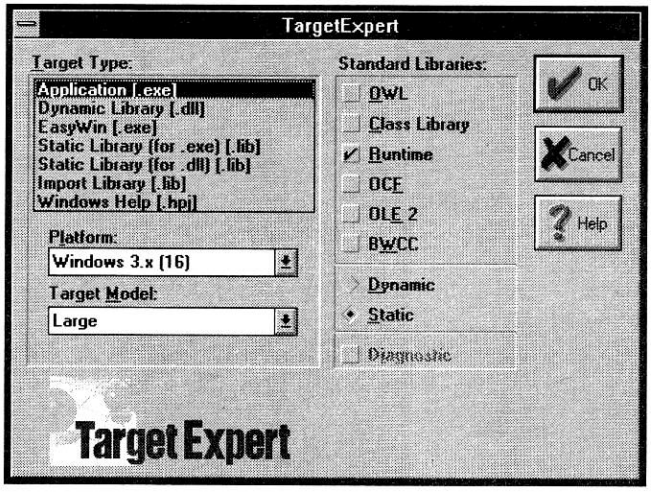
THE PROJECT MANAGER LETS YOU CREATE multiple 16- and 32-bit applications and DLLs from a single project. This gives you the ability to generate both a 16- and a 32-bit application with a single command. For example, you can build two versions of a single application: one that contains debug information, and another that doesn't.

The Project Manager also supports Source Pools to help you organize the files that different targets share. Grouping and sharing core source files in multitargeted projects makes it easy to manage these files as your projects become increasingly large and complex.

Linking the correct libraries

In Help, choose Search All, then Search For "TargetExpert."

TARGETEXPERT SETS UP THE CORRECT LIBRARIES to link, whether you're building a Windows import library, a DOS overlay module, or a 32-bit Windows application. When you use TargetExpert, you never have to remember the name of a single library file.



With TargetExpert, you select the platform, the target type, and the types of libraries you want to use. Then, let TargetExpert configure your project with the appropriate libraries and startup modules.

Setting project options

See the *User's Guide*, Chapter 3.

In Help, choose Search All, then Search For "view options hierarchy."

THE BORLAND C++ IDE HELPS YOU set and modify project options with the Project Options dialog box. Once set, you can quickly view and modify your option settings with the Options Hierarchy dialog box.

The Project Manager lets you set project options for an entire branch in the project hierarchy, and it lets you override the options for any single file in the project. This flexibility gives you the ability to create customized project settings for the most demanding compile and link situations that you will encounter.

In Help, choose Search All, then Search For "Style Sheets."

Also, see STYLESHT.IDE in EXAMPLES\IDE\STYLESHT.

To simplify the task of setting options, the Project Manager provides Style Sheets so you can group option settings into distinct units. By creating several Style Sheets, you can apply a set of options to any node in your project.

For more information...

Automating application development

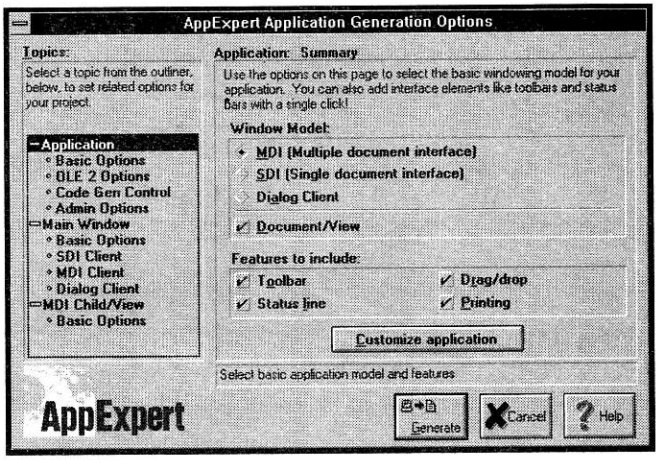
Borland C++ provides the Experts you need to get a jump on your Windows programming tasks. Use AppExpert to generate the source-code foundation of your application, then use ClassExpert to add the functionality that produces a custom application.

Generating Windows applications

See the *User's Guide*, Chapter 4.

In Help, choose Search All, then Search For "AppExpert."

LET APPEXPERT PROVIDE THE SOURCE-CODE FOUNDATION for everything you need in a Windows application. With AppExpert, a click of the mouse brings you the source code for online Help, toolbars, menus, status lines, and many other Windows application features.



In Help, choose Search All, Search For "AppExpert," then "Building Applications with AppExpert."

Instead of coding the foundation of your Windows program yourself, use AppExpert to select the application type, the window features, and the main features of your program. Then click the Generate button and let AppExpert create a complete and working object-oriented Windows application. It saves time and reduces the chance of errors by providing a source-code foundation that you can work with.

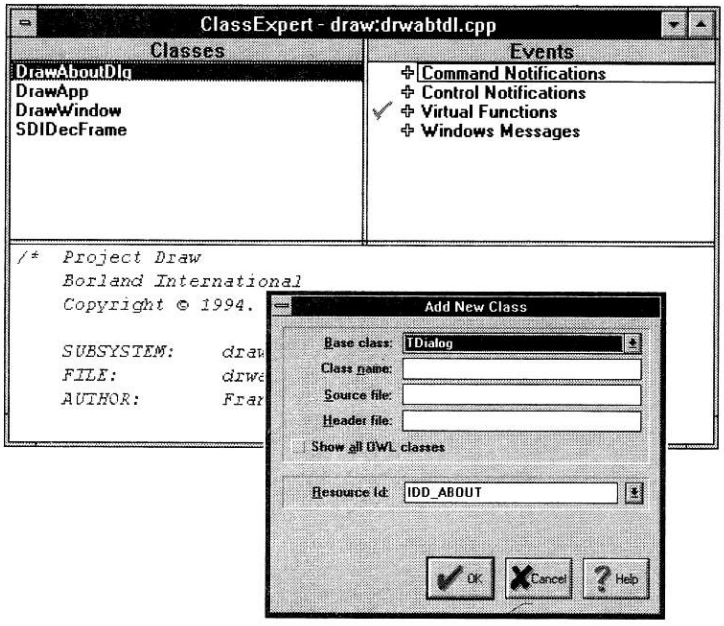
Programming visually

APPEXPERT GETS YOU UP AND RUNNING by generating all the foundation code for your MDI, SDI, and Dialog Client applications. You can create Doc/View applications and applications that support both the container and server models of OLE 2. All this is done by pointing and clicking—you don't need to type a single line of code to create a working ObjectWindows application.

Fine-tuning your AppExpert applications

See the *User's Guide*, Chapter 5. In Help, choose Search All, then Search For "ClassExpert."

USE CLASSEXPERT TO CREATE NEW CLASSES, event handlers, and member functions. You can even use ClassExpert to locate existing classes in your code and to customize the generic functionality of the classes generated by AppExpert.



ClassExpert is the tool you use to modify the source code generated by AppExpert. Click on the event whose functionality you want to enhance, and ClassExpert opens a full-featured editor with the insertion point indicating where to place new code. In fact, because of its BRIEF-like editor and code navigation capabilities, ClassExpert is an ideal environment for developing ObjectWindows programs.

You can use ClassExpert to represent and automate classes in AppExpert projects. You can also tap into existing programs using Rescan to incorporate the code into your new AppExpert applications.

Adding resources to your applications

See the *User's Guide*, Part III.

CLASSEXPERT WORKS SEAMLESSLY WITH RESOURCE WORKSHOP to associate resources with your ClassExpert classes. This lets you visually design the user interface for your AppExpert projects.

For more information...

Designing a graphical user interface

When you're ready to design the user interface for your application, use Resource Workshop. Its graphical environment makes it easy to design user interface components just as they will appear in your application.

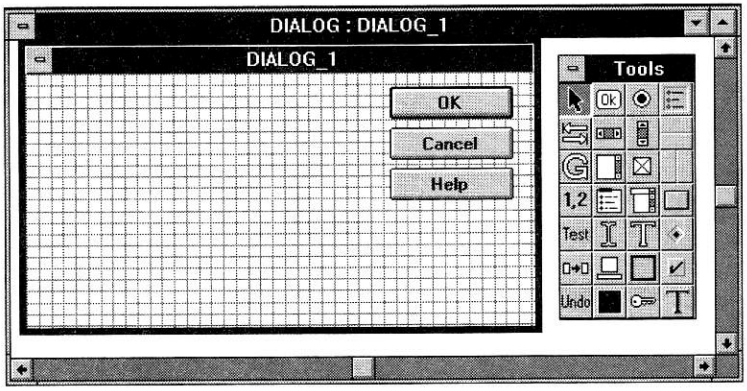
Types of user interface components

See the *User's Guide*, Part III.

RESOURCE WORKSHOP PROVIDES THE TOOLS you need to create and edit a variety of user interface components, including the following:

In Help, choose Search All, then Search For "resources."

- **Dialog boxes.** Create dialog boxes that pop up when a menu command is chosen or when an event occurs in an application. Or, create a dialog box that serves as the main window of an application.
- **Menus.** Create standard menus that appear on the menu bar of an application, or floating menus that appear anywhere on the screen.
- **Accelerators.** Create key combinations that users can press to perform actions.
- **Strings.** Create text strings that contain descriptions, prompts, or error messages.
- **Graphics.** Create bitmaps (.BMP files), icons (.ICO files), cursors (.CUR files), and raster fonts (.FNT files).



Linking user interface components to code

See the *User's Guide*, Chapter 13.

WHEN YOU'RE READY TO LINK user interface components to code, create a resource script (.RC) file using Resource Workshop. Use the IDE or command-line tools to compile the resource script into a binary resource (.RES) file, then link the .RES file to an application. Using ClassExpert, you can automatically generate OWL classes that correspond to user interface components.

For more information...

Using frameworks and libraries

See the *ObjectWindows Programmer's Guide* and the *Library Reference*.

Frameworks and run-time libraries support a wide variety of computational tasks. Each framework is an object-oriented collection of logically related C++ classes and routines designed to solve a particular programming need. The frameworks with Borland C++ provide a solid foundation upon which you can build your application. From within your application, you can then call upon the necessary libraries, which are collections of classes, routines, and functions that perform a wide variety of tasks.

Professional applications with ObjectWindows

See the *ObjectWindows Programmer's Guide*, Chapter 2. In Help, click OWL and select Tasks.

OBJECTWINDOWS IS AN EASY-TO-USE, yet powerful and flexible, object-oriented C++ framework that encapsulates complex windowing functionality. ObjectWindows provides several features designed to help you develop applications that run faster, write code that's easier to debug, and create programs that implement linking and embedding technology. You can use ObjectWindows to develop both 16- and 32-bit applications.

Want to see an ObjectWindows application at work? Take a look at the Borland C++ IDE. It was written using ObjectWindows.

Adding event handlers to your application

See the *ObjectWindows Programmer's Guide*, Chapter 2, and the *ObjectWindows Reference Guide*.

In Help, choose Search All, then Search For "event handlers."

To add event handlers and messages to your application, use ObjectWindows response tables, which map Windows messages to the correct C++ member functions. You can also gain the functionality of event handlers derived from more than one class.

Handling exceptions

See the *ObjectWindows Programmer's Guide*, Chapter 6, and the *ObjectWindows Reference Guide*.

To trap errors that might unexpectedly occur in your program, use ObjectWindows exception-handling objects. ObjectWindows has built-in exception-handling classes that treat exceptions in a consistent, predictable manner.

Handling documents and views

See the *ObjectWindows Programmer's Guide*, Chapter 10.

In Help, choose Search All, then Search For "Doc/View."

You can easily create applications based on the abstract Doc/View model of data handling and display functions. ObjectWindows separates the data from its presentation so you can create multiple views of the same data. When the data is changed in any one of the views, all of the views are updated automatically.

Doc/View applications are flexible. With just a few modifications, you can transform a Doc/View application into an OLE container or server.

Diagnostics

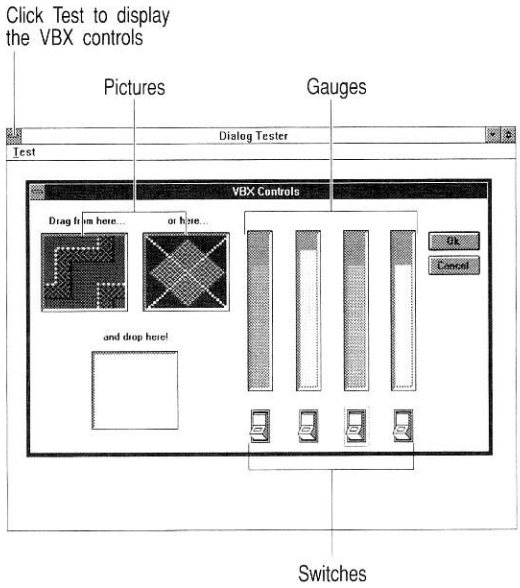
For greater programming safety and increased debugging support, ObjectWindows includes several internal diagnostic classes. You can use these diagnostic classes to catch usage errors or incorrect computations. Built-in diagnostic messages tell you if you have application, window, Doc/View, or GDI-related errors.

Configuring controls and windows

See the *ObjectWindows Programmer's Guide*, Chapter 11, and the *ObjectWindows Reference Guide*.

In Help, choose Search All, then Search For "VBX."

You can incorporate VBX controls from a wide variety of vendors into your 16- and 32-bit applications. ObjectWindows handles VBX controls just like standard ObjectWindows controls. It also provides drag-and-drop support for VBX controls.



The rich set of toolbar, gadget, and dialog classes in ObjectWindows reduces the amount of code you have to write to create sophisticated user interfaces. Toolbar and gadget classes let you customize your user-interface objects. You can easily handle dialog boxes and a dialog box's child controls in an object-oriented manner.

See the *ObjectWindows Programmer's Guide*, Chapter 12.

In Help, choose Search All, then Search For "layout windows."

Just as you can place toolbars and palettes with respect to their parent windows, you can also position other windows in relation to parent windows. Specialized window classes, such as layout windows, let you design constraint-driven windows whose size and position vary as the controlling window's parameters change.

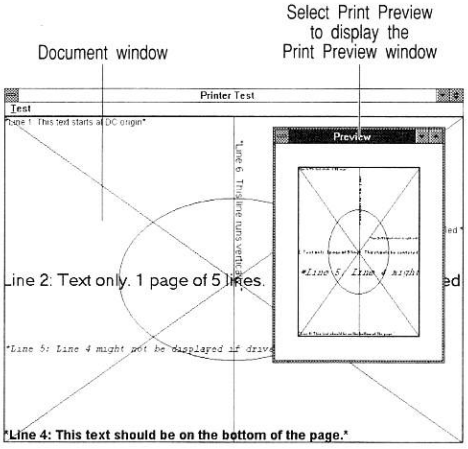
Using frameworks and libraries

Enhancing your application

See the *ObjectWindows Programmer's Guide*, Chapter 12.

In Help, click OWL or choose Search All, then Search For "Clipboard."

By using ObjectWindows clipboard classes, you can browse the contents of the Clipboard and copy data to, and from, the Clipboard. You can use ObjectWindows specialized printer classes to add printer support with print previewing to an application regardless of the type of information being displayed.


Select Print Preview to display the Print Preview window
Document window
Printer Test
Test
Preview
Line 2: Text only. 1 page of 5 lines.
Line 4: This text should be on the bottom of the page.

OLE 2 encapsulation

See the *ObjectWindows Reference Guide*, Part II.

In Help, choose Search All, then Search For "OLE."

You can use the standard ObjectWindows OLE classes by themselves so your application can take advantage of OLE 2 features, such as in-place editing, embedding and linking, and menu merging.

Linking and embedding with ObjectComponents

See the *ObjectWindows Programmer's Guide*, Chapter 18, and the *ObjectWindows Reference Guide*, Part II.

In Help, click OCF.

See the *ObjectWindows Programmer's Guide*, Chapters 19–21, sample source code, and the *ObjectWindows Tutorial*, Chapters 14 and 15.

USE THE OBJECTCOMPONENTS FRAMEWORK to write portable OLE 2-enabled programs. ObjectComponents is a set of classes for creating OLE 2 applications in C++. You can even use ObjectComponents to add OLE 2 support to an existing C++ application.

With ObjectComponents, you can create applications containing spreadsheets from Quattro® Pro, text from Microsoft® Word or WordPerfect,® or objects exported from OLE servers, such as Paradox® or Visio® ObjectComponents also makes it easy to automate your application, so a server application can accept commands from other applications, and a controller application can control the behavior of other applications.

See the *ObjectWindows Programmer's Guide*, Chapters 19–21, and the *ObjectWindows Reference Guide*, Chapter 6.

In Help, click OCF and select Tasks.

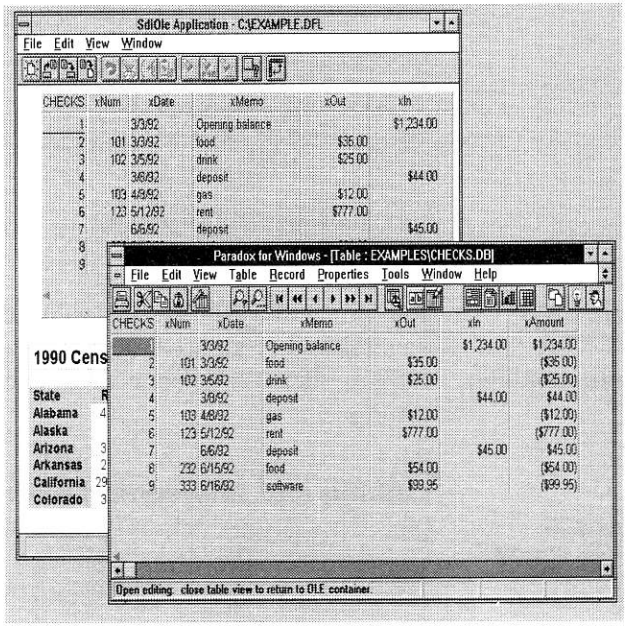
ObjectComponents and OLE 2 compatibility

Because ObjectComponents defines standard user interface features, you can easily add OLE functionality to existing C++ applications. In fact, ObjectComponents applications are compatible with other OLE applications, including OLE 1 applications, even if they weren't built using Borland tools.

In Help, click OCF.

Linking and embedding objects

Using ObjectComponents classes, you can link and embed data from one application into another. To simplify this task, ObjectComponents provides default handlers that cut and paste objects onto the Clipboard. You can easily edit objects in place and merge server and container menus into one.


SdiOle Application - C:\EXAMPLE.DFL
File Edit View Window
Paradox for Windows - [Table : EXAMPLES\CHECKS.DB]
File Edit View Table Record Properties Tools Window Help
Open editing: close table view to return to OLE container.

Create OLE 2 applications using the Experts

See "Automating application development" in this booklet.

To further simplify the process of creating OLE 2-enabled applications, use the Borland AppExpert and ClassExpert, which understand the new ObjectComponents classes.

Automation

See the *ObjectWindows Programmer's Guide*, Chapter 22, and the *ObjectWindows Reference Guide*, Part II.

Add automation to your program so it can talk to other programs. ObjectComponents automates the C++ classes in your application without making structural changes to the classes themselves. Once your program is automated, other programs can talk to it through OLE commands.

Registration and translation

See the *ObjectWindows Programmer's Guide*, Chapters 19–21, and the *ObjectWindows Reference Guide*, Chapter 6.

Because of the flexible, yet powerful, ObjectComponents framework, many complex OLE transactions are transparent to you, the developer. For example, ObjectComponents registers an application with the OLE registry and run-time system. Using the localized information you provide, ObjectComponents looks up commands and descriptions in whatever language you request.

Using the run-time libraries

See the *Library Reference*, Chapter 3, and the *Programmer's Guide*.

In Help, choose Search All, then Search For "run-time libraries."

BORLAND C AND C++ RUN-TIME LIBRARIES are collections of classes, functions, and macros that you can call from within your C and C++ programs to perform a wide variety of processing tasks. These tasks include single thread and multithread processing, mathematical processing, floating-point processing and coprocessor emulation, exception handling, console and GUI initialization, and debugging.

Rather than having to choose the correct run-time files for an application, you can rely on the Borland C++ compiler driver to choose the appropriate libraries for you. If you select compiler options that change the default behavior, the compiler automatically uses the correct libraries.

To give you complete freedom to design applications that work exactly the way you need, Borland provides the source code used in most of the run-time libraries. The many available makefiles help you compile your source code.

Using predefined abstract data types

See the *Library Reference*, Chapter 3.

In Help, choose Search All, then Search For "templates."

THE BORLAND CONTAINER CLASS LIBRARY is a collection of C++ template classes that implements commonly used data structures, such as arrays, lists, iterators, stacks, and queues. The template implementation allows a container to hold virtually any data type. Instead of having to write your own routines, you can call upon these structures to manage data objects you frequently use in your program.

The container classes allow flexibility in how you implement particular data structures. For example, by just changing a type definition, you can switch your stack implementation from a vector-based stack to a list-based stack. All member functions of the stack template class remain the same regardless of the implementation.

For more information...

Debugging and profiling

Borland C++ provides a collection of debuggers and profilers designed to help you fine-tune your applications. Use the various debuggers to locate and eliminate errors from your program code. Use Turbo Profiler to discover how you can improve program speed and performance.

Debuggers

See the *Turbo Debugger User's Guide*, Chapter 3, for a debugging overview.

WHETHER A LOGIC ERROR or a run-time error, you can use the Borland debuggers to help you quickly find the problem.

In Help, choose Search All, then Search For "debugger."

Borland C++ supplies several types of debuggers: one that's integrated into the IDE and others that can be run from within Windows or from the command line.

The integrated debugger

See the *User's Guide*, Chapter 7.

Use the integrated debugger to debug 16-bit Windows programs from the IDE. Using the Edit window as the base for your debugging sessions, you can step through your program line by line, set breakpoints, and view the values of variables as your program runs. You can even modify variables to see how different values affect program behavior.

Standalone Turbo Debugger

See the *Turbo Debugger User's Guide*.

Turbo Debugger is a set of debugging tools that supplements the integrated debugger. With Turbo Debugger, you get three different debuggers: Turbo Debugger for Windows, Turbo Debugger for Win32, and Turbo Debugger for DOS.

Although each version of Turbo Debugger supports a different platform, all three debuggers are almost identical in behavior. You can use Turbo Debugger to set breakpoints and watchpoints, step through code, examine the values of expressions, and modify expression values as the program runs. In addition, you can do the following:

- Set Windows message breakpoints
- Perform Just-In-Time debugging (with TD32)
- Trap C++ exceptions
- Trap operating-system exceptions
- Debug your programs at the assembly level
- Debug multithreaded programs
- Debug your program from a remote machine (either from a network or through serial ports)

See the online file TD_RDME.TXT for information on Just-In-Time debugging.

WinSpector

See the *User's Guide*, Chapter 12.

DOES YOUR PROGRAM TERMINATE with an Unrecoverable Application Error (UAE) or a General Protection (GP) fault? WinSpector and its utilities are debugging tools that help you perform postmortem examinations of Windows programs that end abruptly with a UAE or GP fault. When a UAE or GP fault occurs, WinSpector creates a log file to help you uncover the error.

Pre-processing utilities

In Help, choose Search All, then Search For "WinSpector."

WinSpector includes three pre-processing utilities to enhance the UAE report it creates: BUILDSYM, TMAPSYM, and EXEMAP. With these utilities, you can generate a .MAP file that's useful when source code for the executable file is not available. Once you have a .MAP file, you can create a .SYM file that makes public functions, variable names, and functions in the entry table of the executable available to WinSpector.

Post-processing utility

WinSpector also has a post-processing utility, DFA, that utilizes Turbo Debugger symbolic information to further enhance the readability of the UAE reports that it creates.

WinSight

See the *User's Guide*, Chapter 12.

In Help, choose Search All, then Search For "WinSight."

WINSIGHT GATHERS INFORMATION about window classes, application windows, and window messages. With WinSight, you can study how an application creates its classes and windows, and monitor how different windows send and receive messages.

WinSight is a passive observer—it displays information about messages, but does not prevent messages from getting posted to the appropriate applications. Use WinSight first to find the windows you want to monitor, then to trace the messages that get sent.

Turbo Profiler

See the *Turbo Profiler User's Guide*.

USING THE STATISTICAL REPORTS generated by Turbo Profiler, you can discover the elusive bottlenecks that sap the speed of your 16-bit programs. You can then fine-tune your programs and improve performance in an informed, controlled manner.

In addition to using Turbo Profiler to improve performance, you can use Turbo Profiler to perform coverage analysis, which detects if and when the routines in your program get executed.

Building DOS applications

Borland C++ provides tools to develop new or upgrade existing DOS applications. BCC, the 16-bit compiler, creates 16-bit DOS real-mode applications. BCC32, the 32-bit compiler, creates 32-bit applications that run in console mode on Windows NT or Windows 95.

You can also develop DOS applications from the IDE by using the TargetExpert.

See the *DOS Reference*, Chapter 1.

If your applications are already using all available conventional memory, you can continue to add to your program by using overlays. Large overlaid programs can exceed the 640K DOS memory limitation by swapping code segments to the disk.

Developing protected-mode applications

If you ordered PowerPack, see the *PowerPack for DOS User's Guide*.

THE OPTIONAL BORLAND POWERPACK for DOS gives you the capacity for developing DOS protected-mode applications that break the 640K DOS memory limitation by using extended memory. It also gives your DOS applications the highest performance possible. PowerPack provides you with the following:

- Ability to share DLLs between DOS and Windows applications
- 16- and 32-bit DOS dynamic-link libraries
- 32-bit flat memory
- Access up to four Gigabytes of available memory
- Virtual memory management
- 16- and 32-bit DPMI version of the Borland Graphics Interface
- Turbo Vision 2.0, a 16- and 32-bit DOS framework

Building applications with Turbo Vision

If you ordered PowerPack, see the *Turbo Vision Programmer's Guide*.

TURBO VISION 2.0 IS INCLUDED IN THE optional PowerPack for DOS. Turbo Vision is a complete object-oriented framework that offers all the basic functionality you need for creating windowed DOS applications. Turbo Vision includes the following:

- Multiple, resizeable, overlapping windows
- Pull-down menus
- Mouse support
- Dialog boxes
- Built-in color installation
- Buttons, scroll bars, input boxes, check boxes, and radio buttons

Using Turbo Vision gives all your applications a consistent state-of-the-art look and feel with minimal effort on your part.

For more information...

Working from the command line

When you work in the IDE, much of the guesswork of selecting the right compiler and options is done for you. But if you prefer to work from the command line, Borland C++ provides fast, direct-access tools for compiling, linking, debugging, and managing projects. For fast execution, these tools run as protected-mode applications built with the optional Borland PowerPack for DOS.

Using the compilers and linkers

See the *User's Guide*, Part II. Look at the online file README.TXT for the latest details.

BORLAND C++ PROVIDES YOU WITH 16- and 32-bit command-line compilers and linkers that let you develop your DOS and Windows programs from the command line. With the 32-bit command-line tools, you can also build 32-bit GUI applications and applications for Win32s and Windows NT.

In Help, choose Search All, then Search For "command-line compiler."

The enhanced 16-bit linker provides post-link optimization that removes redundant information in executable files. File size is smaller and application startup time is reduced.

Using resource tools

RESOURCE COMPILERS, LINKERS, AND SHELLS are also available from the command line:

In Help, choose Search All, then Search For "BRCC," "RLINK," "BRC," or "resource tools."

See the *User's Guide*, Chapter 8.

- **Resource Compilers** compile resource script files and produce binary .RES files.
- **Resource Linkers** bind resources, in .RES file form, to an executable file and mark the resulting file as a Windows executable.
- **Resource Shells** are shells that let you start both BRCC or BRCC32 and RLINK or RLINK32 in a single step.

Managing your project with MAKE

See the *User's Guide*, Chapter 10.

TO FREE YOURSELF from having to track how your project gets built, you can write project information into a file. By writing into this file the conditions under which each part of your project is built, you can let your system and the MAKE utility decide when to update your project.

You define the rules MAKE uses to determine which files to build and how to build them. Your instructions can be as complex or as simple as needed to accomplish your programming needs.

For more information...

C and C++ languages

More than any other compiler, Borland C++ gives you easy access to the programming power of the evolving C++ language.

When you're developing cutting-edge applications, you need a programming language that's as easy as possible to use. To design high-quality applications, your programming language must be simultaneously hardware independent yet capable of exploiting all machine resources.

Borland C++ gives you exactly the kind of language support you need. The world leader in C and C++ implementation gives you industry standards, forward-looking technology, and compatibility.

ANSI C

See the *Programmer's Guide*, Appendix A. In Help, choose Search All, then Search For "ANSI."

BORLAND C++ COMPILERS ARE FULLY compliant with the ANSI C X3J11/88-159 Standard. Only Borland C++ is certified by BSI/ISO for C compliance.

Any C program that complies with the ANSI Standard can be compiled with Borland C++ compilers. Also, a program compiled under Borland C++ with language extensions disabled is highly portable to other ANSI C compilers.

C++

In Help, choose Search All, then Search For "C++ Options."

BORLAND CLOSELY TRACKS the evolving ANSI/ISO committee definition of the C++ language. Just a few C++ features that Borland C++ implements include:

See the *Programmer's Guide*.

- **Templates** develop generic classes and generic functions that accept parameterized types.
- **Exception handling** provides a standard method to handle run-time errors. Only Borland C++ provides a way to mix structured exceptions developed in C with the C++ code.
- **New-style type-safe casting** makes safe type conversions even when you're working with polymorphous classes.
- **Run-time type identification (RTTI)** determines the actual type of a data object at run time, including virtual, reference, and pointer types.
- **Three distinct character types** overload functions based on a distinction between **char**, **signed char**, and **unsigned char.**
- **Parameterized manipulators based on templates** develop generic I/O streams formats that can accept any parameter type.

C and C++ languages

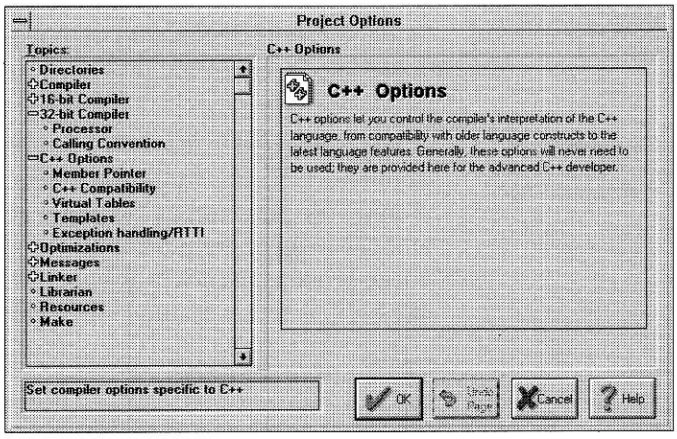
For more information...

See the *User's Guide*, Chapter 3, and the *Borland C++ Quick Reference card.*

In Help, choose Search All, then Search For "C++ Options."

You can selectively apply many C++ language constructs by using compiler options and choosing the appropriate libraries. For example, you can eliminate exception-handling from the run-time libraries by linking with the NOEH libraries. You can also use a compiler option to avoid generating run-time type identification. If three distinct character types are a problem with existing code, you can use a compiler option and the appropriate header files to revert to the old behavior.

The Project Options dialog box lets you set C++ options.



Language extensions

BORLAND C++ EXTENDS THE LANGUAGE definition by adding keywords that are not part of either the ANSI C Standard or the C++ language. These keywords are options that work within the syntax of the C and C++ languages.

See the *Programmer's Guide*, Chapters 1 and 4.

Because the C and C++ languages are machine independent, your machine's resources might not be used efficiently. Many of the language extensions are provided specifically so that you can use machine resources more efficiently. For example, the **__try** and **__finally** keywords are C language extensions that can handle system signals to indicate program exceptions. You can also use **__far**, **__near**, and **__huge** to make better use of segmented memory.

You can use a compiler option to disable all extensions and accept only code that complies with the ANSI Standard.

For more information...

Program portability

Borland C++ provides tools that can target multiple platforms from a single environment. Borland C++ supports all DOS and Windows operating systems and APIs based on Intel PC processors. DOS applications address these processors in real mode; Windows 3.x applications address them in protected mode. With the optional Borland PowerPack for DOS, you can also develop 16- and 32-bit DOS protected-mode applications.

Migrating applications

In Help, choose Search All, then Search For "DOS applications."

BORLAND C++ PROVIDES MIGRATION PATHS across platforms. Borland C++ is designed to minimize the amount of work required to update DOS or Windows applications for use with one of the newer versions of DOS or Windows.

See the *User's Guide*, Appendix A. In Help, choose Search All, then Search For "EasyWin."

There is no need to leave your existing DOS code behind. EasyWin simulates DOS in a window that lets you run standard input/output C and C++ programs as Windows applications. Write handy utilities without the programming overhead of *WinMain* and message loops. EasyWin is a great way to experiment with state-of-the-art C and C++ features supported by Borland C++.

DOS applications

Real-mode DOS applications developed with Borland C++ don't need to be recompiled to run as a console-mode application on Windows NT or Windows 95:

- Most real-mode DOS applications can be recompiled without modifications to run as an EasyWin application on Windows 3.x.

Ask your dealer for Borland PowerPack for DOS.

- Most 32-bit DOS applications developed with the PowerPack for DOS can run under DOS, Windows NT, a Windows 3.1 or 3.11 DOS shell, or a OS/2 2.x DOS shell.
- Most DOS dynamic-link libraries (16- or 32-bit) can be used in a 16- or 32-bit Windows application, respectively, without recompiling.

Windows applications

Borland simplifies migrating from one Windows version to the next. With Borland C++, you can:

See the *ObjectWindows Programmer's Guide*, Chapters 19–21.

- Add OLE now. The ObjectComponents Framework adds OLE with minimal changes to your code.

In Help, click OWL or OCF.

- Change ObjectWindows applications from 16-bit Windows to 32-bit Windows for Windows NT or Windows 95 with the flip of a switch. Just rebuild and run.

For more information...

See the *User's Guide*, Part III. In Help, choose Search All, then Search For "VBX."

- Take your VBX controls with you. Even though VBX controls are 16-bit technology, Borland lets you plug them into 32-bit programs including programs running on Windows NT.

Developing international applications

TO DISTRIBUTE YOUR APPLICATIONS to more than one locale, Borland C++ provides support for international software development. In addition to the default locale, you can link run-time library support for the following:

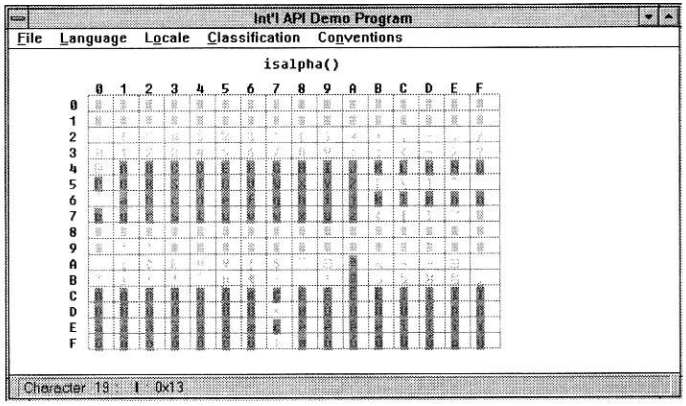
See the *Library Reference*, Chapter 1.

- German
- French
- English (Great Britain)
- English (United States)

In Help, choose Search All, then Search For "locale" and "language."

The results of selecting a locale can be seen by using the INTLDEMO example. The example shows you what characters are processed as alphabetical, digit, space, and so on, according to the ANSI functions. The example also shows how locale specification determines the sort order of file names.

See the complete example in EXAMPLES\OWL\OWLAPPS\INTLDEMO. See also DOC\INTLDEMO.TXT.



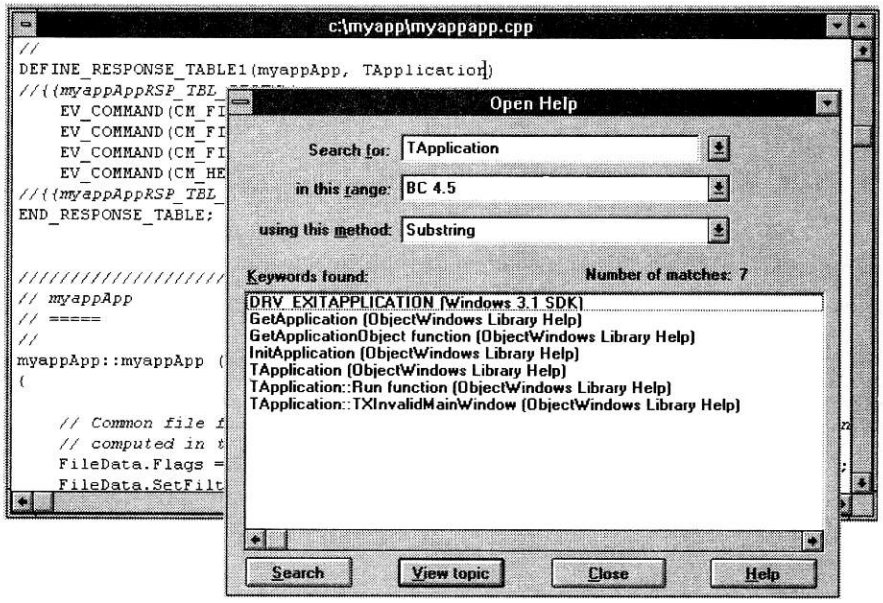
Learning more about Borland C++

Information on Borland C++ is available in many forms to help you learn more about the product.

Making online Help work for you

Choose Help from the IDE main menu or press F1 anywhere in the IDE.

BORLAND C++ ONLINE HELP offers an abundance of programming assistance. Whether you need Help on a class or function, want to figure out what an error message means, or are curious about a dialog box setting, you can press F1 for instant information from the IDE. You can even get Help on both the 16- and 32-bit Windows APIs.



See the *User's Guide*, Part II. In Help, choose Search All, then Search For "OpenHelp."

OpenHelp—Borland's unique extension to Windows Help—lets you search for topics in multiple Help files, including ones you create yourself. By customizing OpenHelp, you can define search ranges to locate topics specific to Windows, ObjectWindows, DOS, or in any combination you choose.

Learning by example

Look in the \BC45\EXAMPLES directories online.

OVER 100 SAMPLE PROGRAMS show how to accomplish common programming tasks using Borland tools. Samples are organized in directories online according to topics, such as Windows, ObjectComponents, ObjectWindows, and DOS programming.

In online Help, you can find a myriad of examples of APIs, run-time library functions, and resource scripts, many of which you can cut and paste directly into your application.

Learning more about Borland C++

Viewing documentation online

Click the online manuals icon to find what you need.

THE CD-ROM VERSION OF BORLAND C++ includes the complete documentation set online:

- General programming information
  User's Guide
  Programmer's Guide
  Library Reference
  Class Libraries Guide
  DOS Reference
- Additional programming tools
  Turbo Debugger User's Guide
  Turbo Profiler User's Guide
- ObjectWindows programming
  ObjectWindows Tutorial
  ObjectWindows Programmer's Guide
  ObjectWindows Reference Guide

State-of-the-art viewing and searching technology with the DynaText Browser provides quick access to the information you need. In the online books, you can quickly locate and print the information you need. Searches include multiple keywords, wildcard characters, and proximity searches. Hypertext links let you jump directly to related topics.

Reading Borland C++ printed documentation

Registered users can purchase the Borland C++ books by calling Borland Customer Service at 800-682-9299 (in the US), or contacting your Borland representative.

IF YOU PREFER PRINTED MANUALS, all the Borland C++ books are available in printed form. Together with this *Quick Tour*, the Quick Reference card, and the ObjectWindows Quick Reference poster, you have all you need to develop applications with Borland C++.

Watching videos

Order videos from your Borland representative, or call Customer Service.

IF YOU'D RATHER, you can take a class at your own convenience. Borland's Video Series offers two C++ videos to introduce you to C++ programming:

- *World of C++* —Moving from C to C++
- *Beyond the World of C++* —Intermediate to advanced C++

Getting support

FOR MORE HELP, the Borland Assist program offers technical support plans to suit different users' needs, from individual consultants to large corporations. For details, see the *Borland Assist Support and Services Guide* included with this product.

For more information...

To order these products, contact your Borland representative or call Customer Service at 800-682-9299, ext. 1031 (in the US).

Expanding the power of Borland C++

Borland C++ is the core of a large family of products that together fill the needs of any serious application developer. With additional products, add-ons, and extensions, Borland gives you the tools you need to build robust, state-of-the-art, portable applications for just about any PC platform.

Accessing data with database tools

Ask for Borland C++ and Database Tools to get these tools in the same box with Borland C++.

To get the tools separately, ask for the Borland Database Engine and for SQL Links.

IF YOU WANT TO ADD DATABASE SUPPORT to your Windows applications, Borland provides database tools that work together with Borland C++. These tools allow you to build applications that perform data acquisition, data analysis, and more.

The Borland Database Engine—the core database functionality found in Paradox for Windows, dBASE for Windows, and Quattro Pro for Windows—is now available to C and C++ developers. Use it to access, analyze, and manipulate data stored in those databases. Then add SQL Links to access data on remote servers, such as InterBase, Oracle, Sybase, Informix, and others.

Putting VBX controls in your Windows programs

Ask for the Borland Visual Solutions Pack.

TO TAKE ADVANTAGE OF MANY of the prepackaged VBX controls available today, Borland combined many of the industry's most popular and versatile controls together in one box. The Borland Visual Solutions Pack is a collection of more than 30 drop-in components that you can integrate into your Windows application, whether you've built it with Borland C++, dBASE for Windows, Visual Basic, Visual C++, or any other application development tool that lets you use VBX controls. And only Borland C++ lets you use 16-bit VBX controls in your 32-bit applications.

The Visual Solutions Pack includes controls for ODBC database support, spreadsheets, WYSIWYG word processors, 3-D and 2-D charts, image editors, serial communications interfaces, notebook tabs, animated buttons, gauges, sliders, and a host of fun gadgets.

Developing Windows Help

Ask for the ForeHelp Help authoring tool.

TO SIMPLIFY YOUR DEVELOPMENT OF Windows Help systems, Borland provides ForeHelp. ForeHelp is a Windows development tool that helps you create Windows Help files without having to understand RTF file formats and without having to interact with the Windows Help compiler. ForeHelp provides a visual development environment that features a full-featured WYSIWYG word processor, as well as complete tools to manage your Help projects.

Expanding the power of Borland C++

Developing for OS/2

Ask for Borland C++ for OS/2 with ObjectWindows.

IF YOU DEVELOP DIRECTLY FOR OS/2, OR ARE interested in moving your Windows applications to OS/2, use Borland C++ for OS/2. The most complete C++ development environment for OS/2, the Presentation Manager-hosted IDE includes the tools you need to edit, compile, debug, and run your OS/2 applications quickly. ObjectWindows for OS/2 helps you take your existing ObjectWindows applications to OS/2 with a minimum of effort.

Developing DOS extended applications

Ask for the Borland PowerPack for DOS.

TO TAKE ADVANTAGE OF MEMORY beyond the 640K DOS limit, you can build DOS protected-mode applications with the Borland PowerPack for DOS. The PowerPack plugs into your Borland C++ development environment, allowing you to use the same IDE and tools to build your DPMI applications that you use for your Windows development. You can maintain and enhance existing DOS applications, or begin migrating your DOS applications to Windows. The PowerPack also includes 16- and 32-bit DPMI versions of the Borland Graphical Interface, to simplify the design and creation of fast, clean graphics. And Turbo Vision 2.0 helps you create powerful, windowed and mouse-driven user interfaces for your DOS applications.

Writing low-level routines

Ask for Turbo Assembler.

THE BORLAND TURBO ASSEMBLER IS A MULTI-PASS assembler that runs on IBM PC computers and compatibles and generates instructions for the 8086, 8088, 80286, 80386, i486, and Pentium, along with 8087, 80287, and 387 numeric coprocessors. This fast, multi-pass assembler provides an interface between C, C++, Pascal, FORTRAN, and COBOL; and 32-bit model and stack frame support. You can link 16- and 32-bit Borland C++ modules with Turbo Assembler modules and produce fast, executable files.

Editing your programs

Ask for the BRIEF editor.

THE BRIEF FULL-FEATURED EDITOR PROVIDES you with the features you need to write professional applications productively in DOS. The powerful macro language lets you customize your programming environment, and its SETUP program ensures BRIEF works instantly with your software. BRIEF provides a host of powerful and flexible features including bookmarks, search and replace, column blocking, 300 levels of undo/redo, and more.

Borland may have patents and/or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents.

COPYRIGHT © 1994 Borland International. All rights reserved. All Borland products are trademarks or registered trademarks of Borland International, Inc. Other brand and product names are trademarks or registered trademarks of their respective holders.

1E0R1094
9495969798-9 8 7 6 5 4 3 2 1
H1

Borland

Corporate Headquarters: 100 Borland Way, Scotts Valley, CA 95066-3249, (408) 431-1000. Offices in: Australia, Belgium, Brazil, Canada, Chile, Denmark, France, Germany, Hong Kong, Italy, Japan, Korea, Latin America, Malaysia, Netherlands, Singapore, Spain, Sweden, Taiwan, and United Kingdom • Part # BCP1245WW21770 • BOR 7779